(12) UK Patent Application (19) GB (11) 2 180 128 (13) A

(43) Application published 18 Mar 1987

(21) Application No 8521421

(22) Date of filing 28 Aug 1985

(71) Applicant
Anamartic Limited,
(Incorporated in United Kingdom),
Milton Hall, Milton, Cambridge CB4 4AE

(72) Inventor
Richard J. Westmore

(74) Agent and/or Address for Service
Reddie & Grose, 16 Theobalds Road, London WC1X 8PL

(51) INT CL[4]
G09G 1/02 G11C 7/00

(52) Domestic classification (Edition I)
H4T 120 122 123 125 CJA
G4A MG1

(56) Documents cited
None

(58) Field of search
H4T
G4A
Selected US specifications from IPC sub-classes G06F G09G

(54) **Window graphics system**

(57) The system comprises a plurality, e.g. some tens or hundreds, of microframe stores 10, each having the capacity to store one "pane" of a window in world-space, where a pane occupies a relatively small rectangular area of the display screen, say around 10%. A graphics processor 11 and store controller 12 enable data to be written into the microframe stores and also to send to each store coordinate values defining its position in screen space. Video data is read out from the stores onto a pixel bus 18 synchronously relative to sync pulses on a line 15 so as to place the image provided by each store in the correct location on the screen. The full screen image is synthesized from the contributions from the MF stores which can be assigned arbitrarily to different viewports and different bit planes for the pixel bus 18. Whether or not a store writes onto the pixel bus can be determined by a priority contest, using a priority number passed from store to store on a priority bus 19.

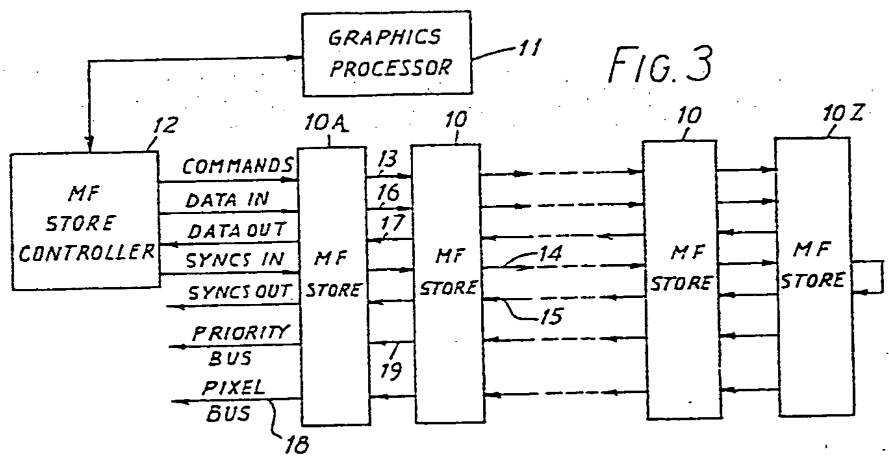


FIG. 3

The drawings originally filed were informal and the print here reproduced is taken from a later filed formal copy.

FIG. 1

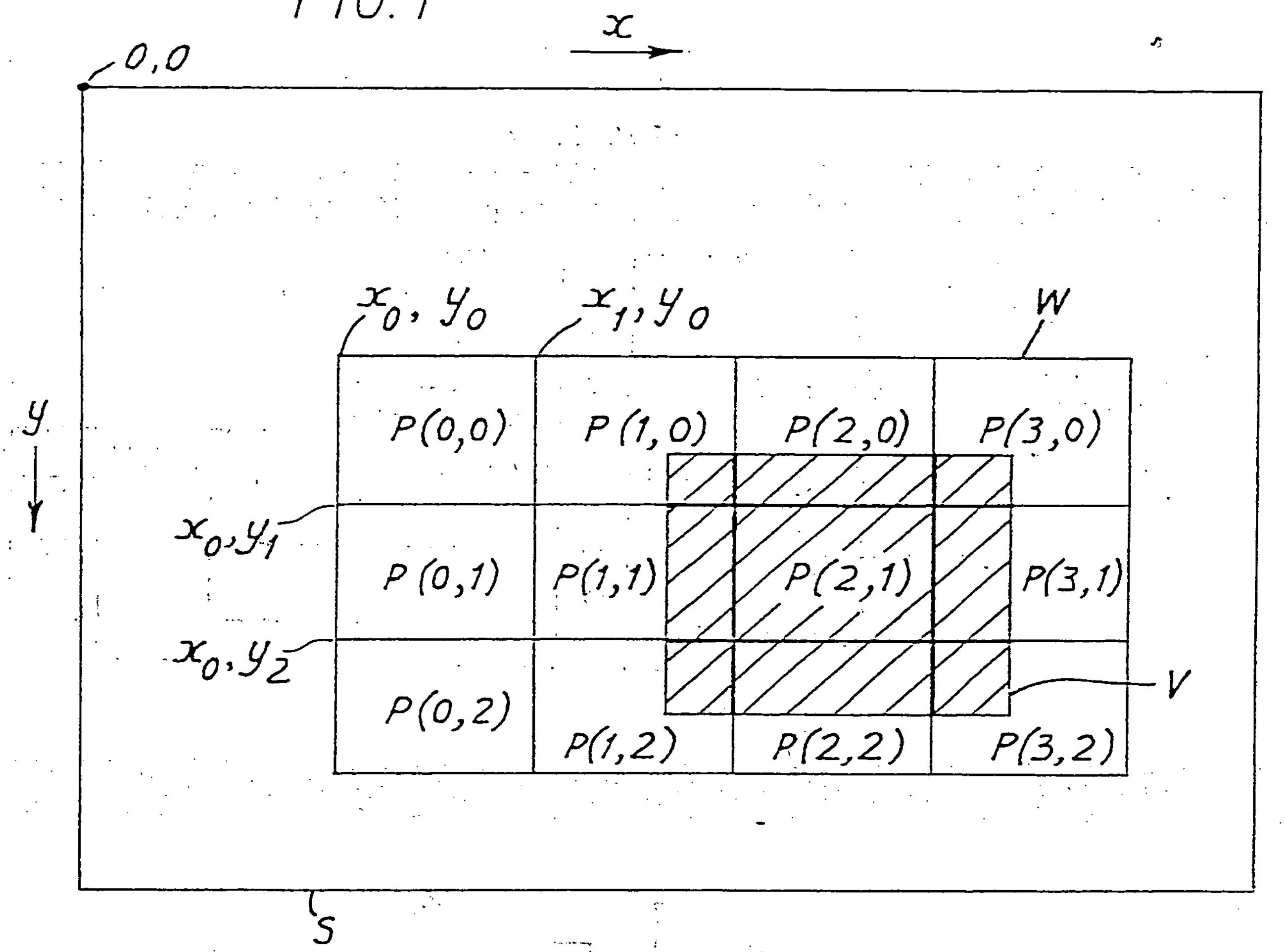
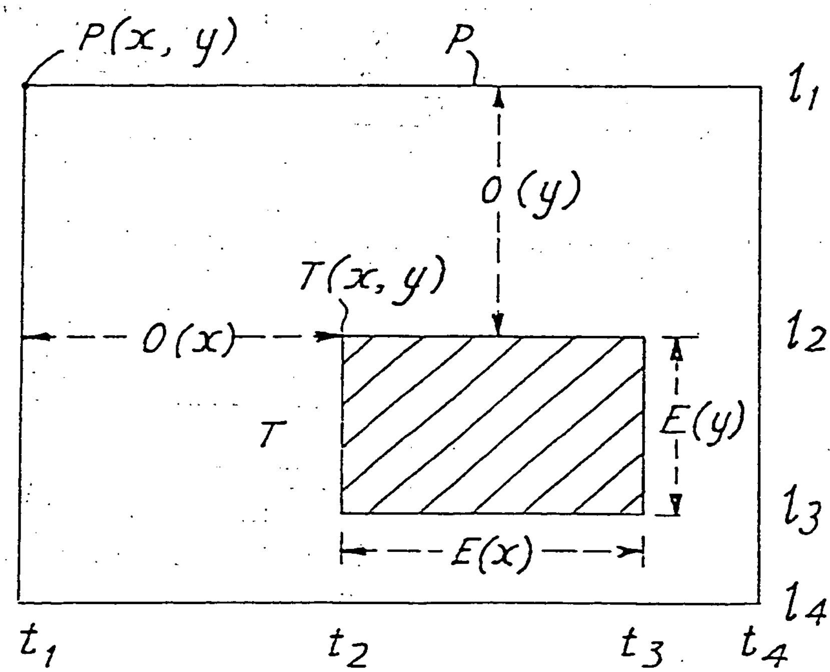

x
0,0
y
x_0, y_0
x_1, y_0
W
P(0,0)
P(1,0)
P(2,0)
P(3,0)
x_0, y_1
P(0,1)
P(1,1)
P(2,1)
P(3,1)
x_0, y_2
P(0,2)
V
P(1,2)
P(2,2)
P(3,2)
S

FIG. 2


P(x, y)
P
l_1
0 (y)
T(x, y)
0 (x)
l_2
T
E(y)
l_3
E(x)
l_4
t_1
t_2
t_3
t_4

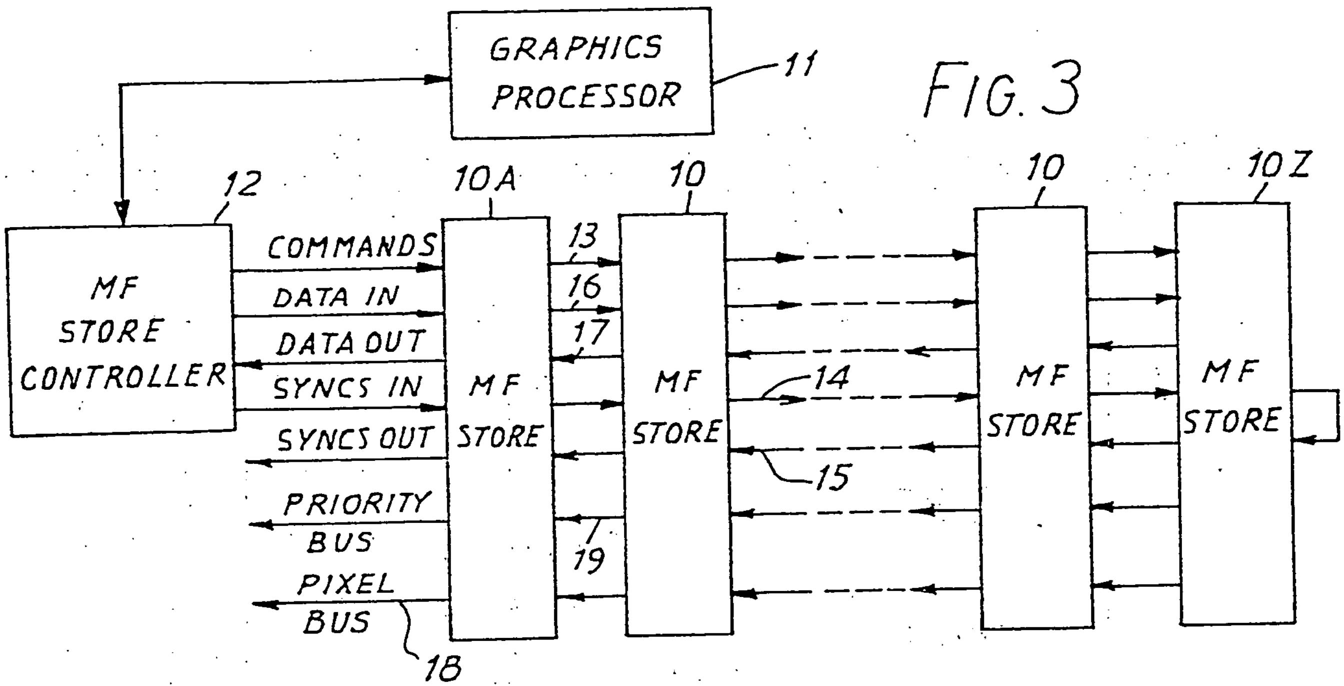

GRAPHICS
PROCESSOR
11
12
10A
10
10
10Z
MF
STORE
CONTROLLER
COMMANDS
DATA IN
DATA OUT
SYNCS IN
SYNCS OUT
PRIORITY
BUS
PIXEL
BUS
18
13
16
17
14
15
19
MF
STORE
MF
STORE
MF
STORE
MF
STORE

FIG. 3

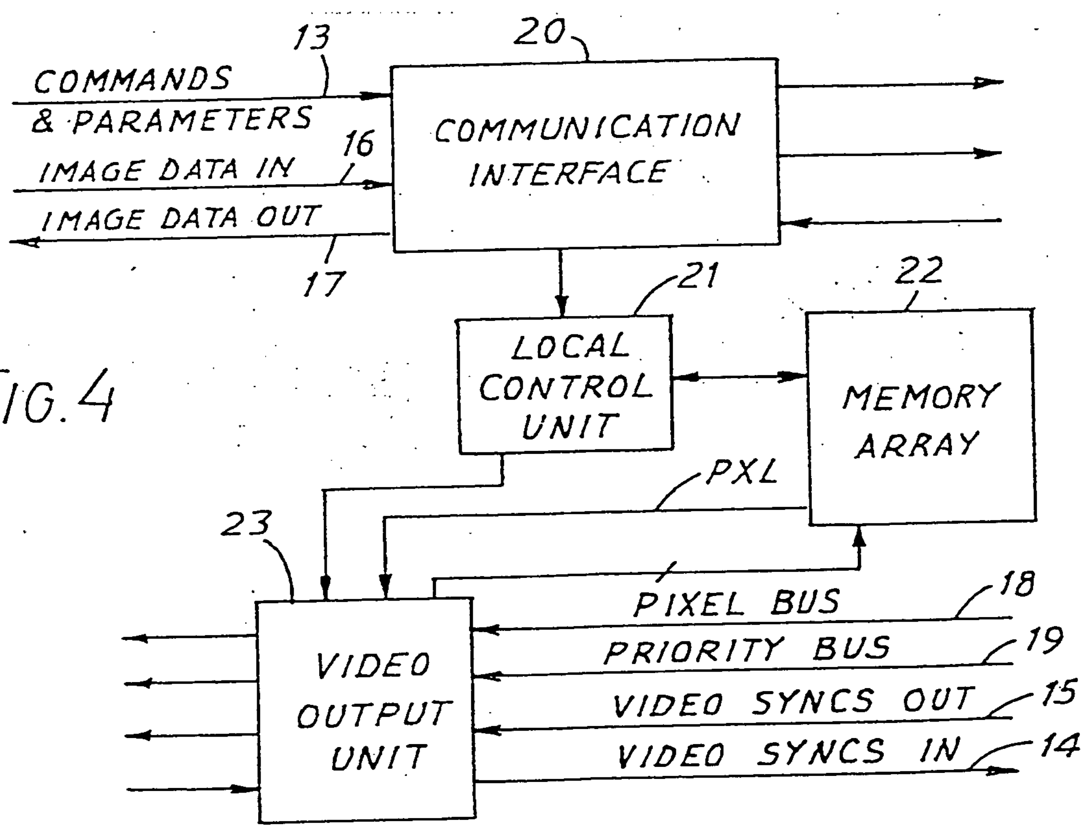
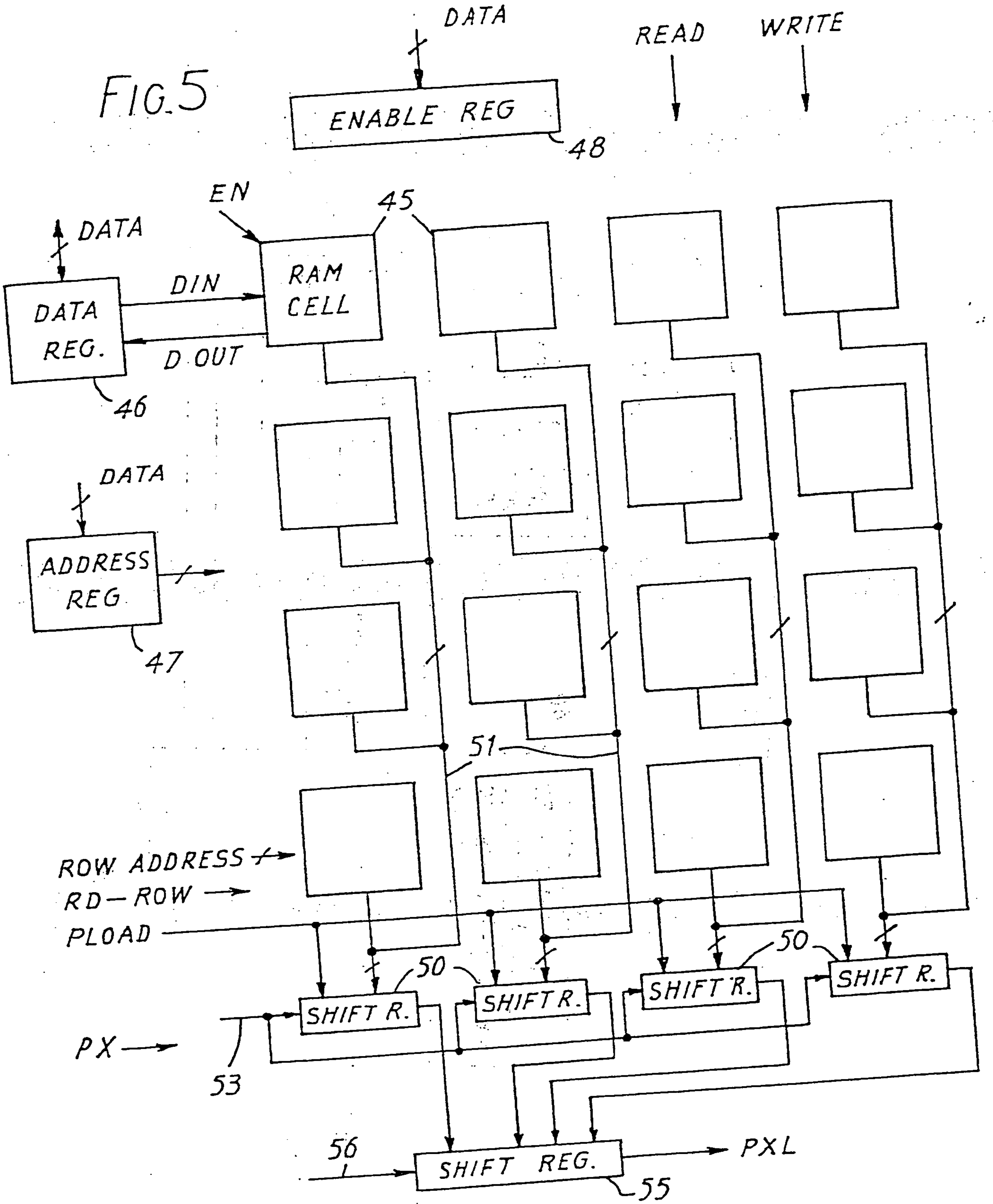

13
20
COMMANDS
& PARAMETERS
16
IMAGE DATA IN
IMAGE DATA OUT
17
COMMUNICATION
INTERFACE
21
22
LOCAL
CONTROL
UNIT
MEMORY
ARRAY
PXL
23
VIDEO
OUTPUT
UNIT
PIXEL BUS
18
PRIORITY BUS
19
VIDEO SYNCS OUT
15
VIDEO SYNCS IN
14

FIG. 4

FIG.5


DATA
READ
WRITE
ENABLE REG
48
EN
45
DATA
DIN
RAM CELL
DATA REG.
D OUT
46
DATA
ADDRESS REG
47
51
ROW ADDRESS
RD-ROW
PLOAD
50
SHIFT R.
SHIFT R.
SHIFT R.
SHIFT R.
PX
53
56
SHIFT REG.
PXL
55

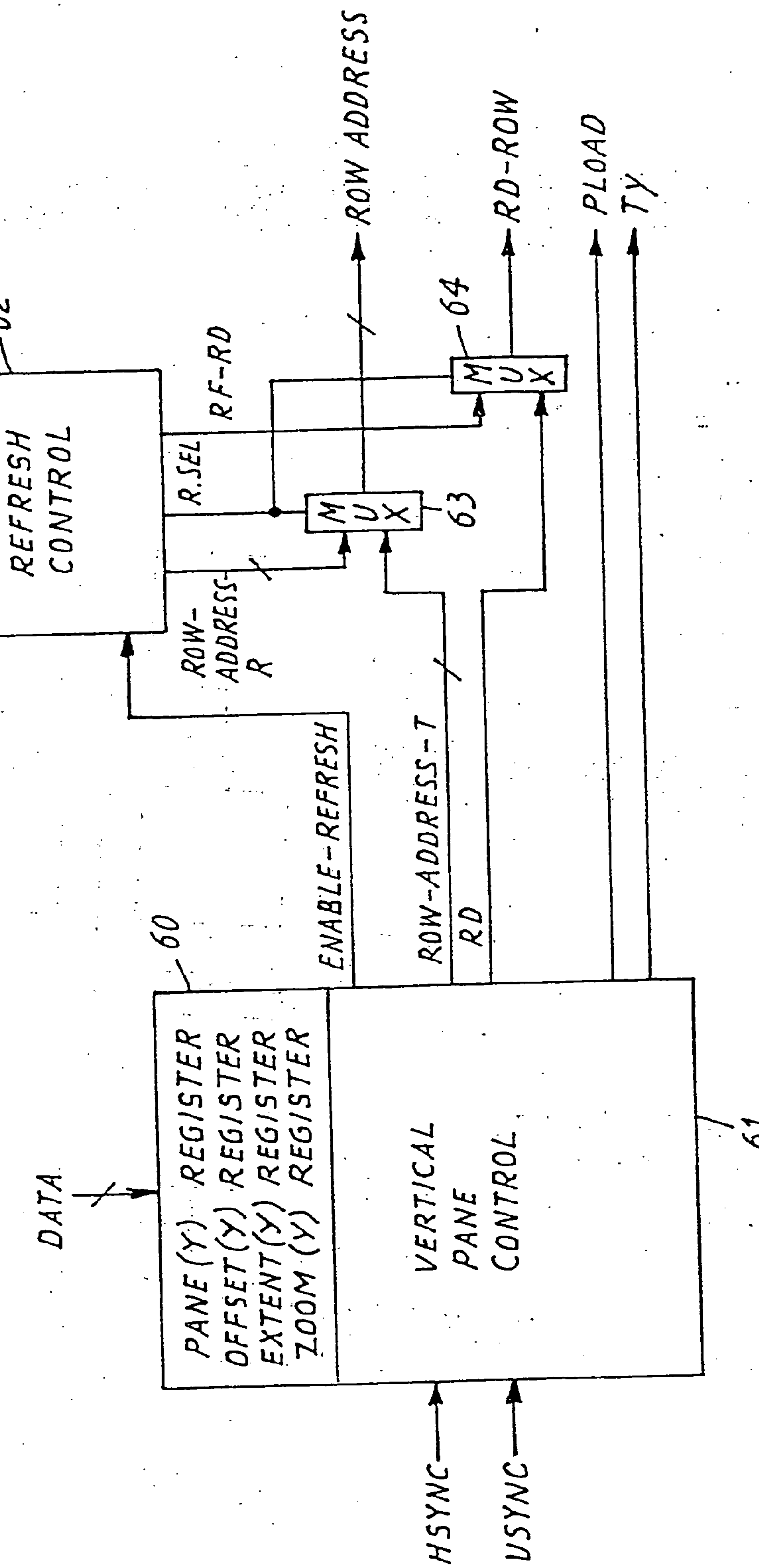
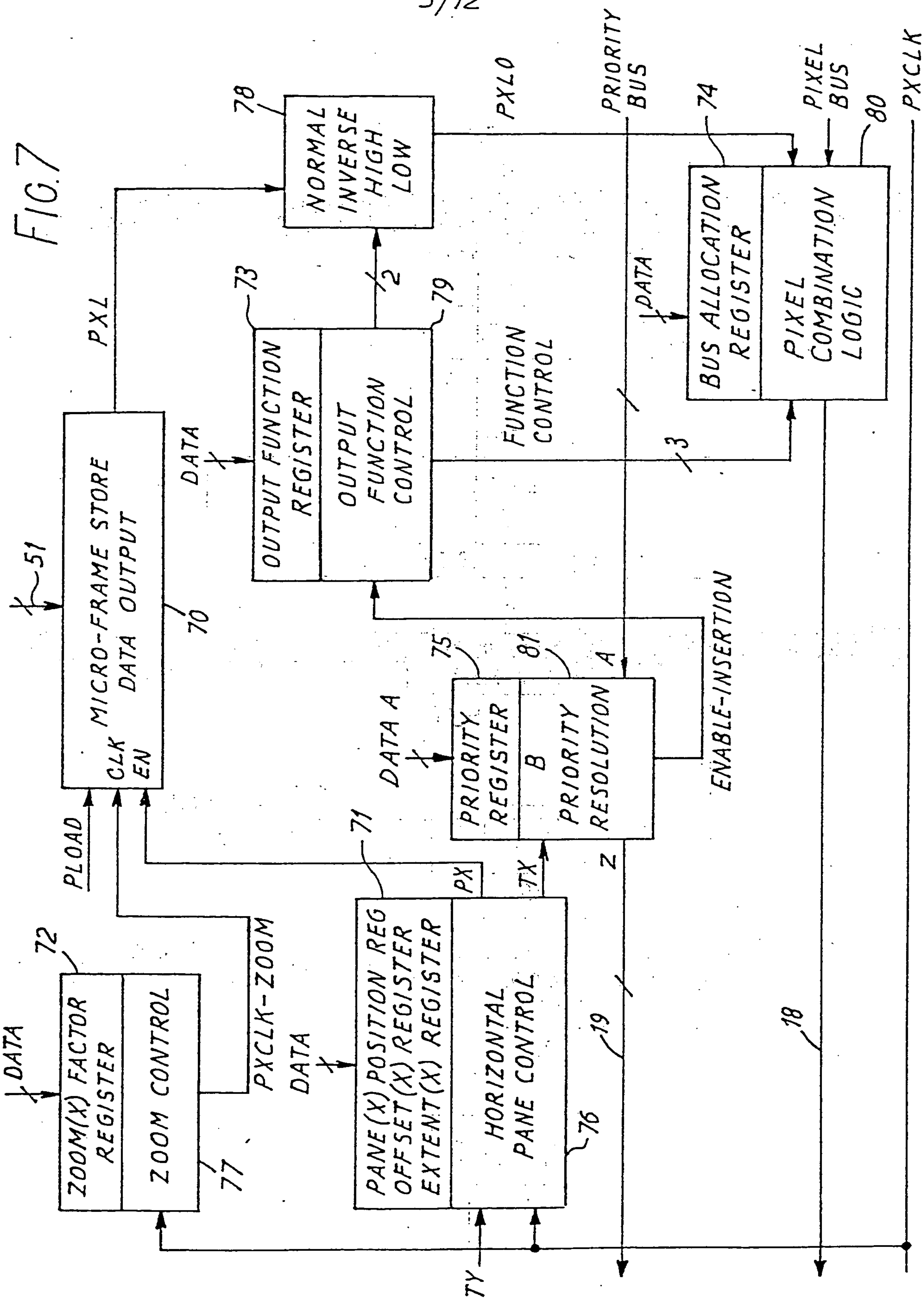

REFRESH CONTROL
62
RF-RD
R.SEL
ROW-ADDRESS-R
ROW ADDRESS
RD-ROW
PLOAD
TY
MUX
64
MUX
63
ENABLE-REFRESH
ROW-ADDRESS-T
RD
60
PANE (Y) REGISTER
OFFSET (Y) REGISTER
EXTENT (Y) REGISTER
ZOOM (Y) REGISTER
DATA
VERTICAL PANE CONTROL
61
HSYNC
VSYNC

FIG. 6

FIG.7


MICRO-FRAME STORE
DATA OUTPUT
CLK
EN
51
70
PLOAD
PXL
NORMAL
INVERSE
HIGH
LOW
78
PXLO
OUTPUT FUNCTION
REGISTER
OUTPUT
FUNCTION
CONTROL
73
79
2
DATA
FUNCTION
CONTROL
3
PRIORITY
BUS
BUS ALLOCATION
REGISTER
PIXEL
COMBINATION
LOGIC
74
80
PIXEL
BUS
PXCLK
DATA
ENABLE-INSERTION
PRIORITY
REGISTER
B
PRIORITY
RESOLUTION
75
81
A
DATA A
Z
ZOOM(X) FACTOR
REGISTER
ZOOM CONTROL
DATA
72
77
PXCLK-ZOOM
PANE(X) POSITION REG
OFFSET(X) REGISTER
EXTENT(X) REGISTER
HORIZONTAL
PANE CONTROL
71
76
DATA
PX
TX
TY
19
18

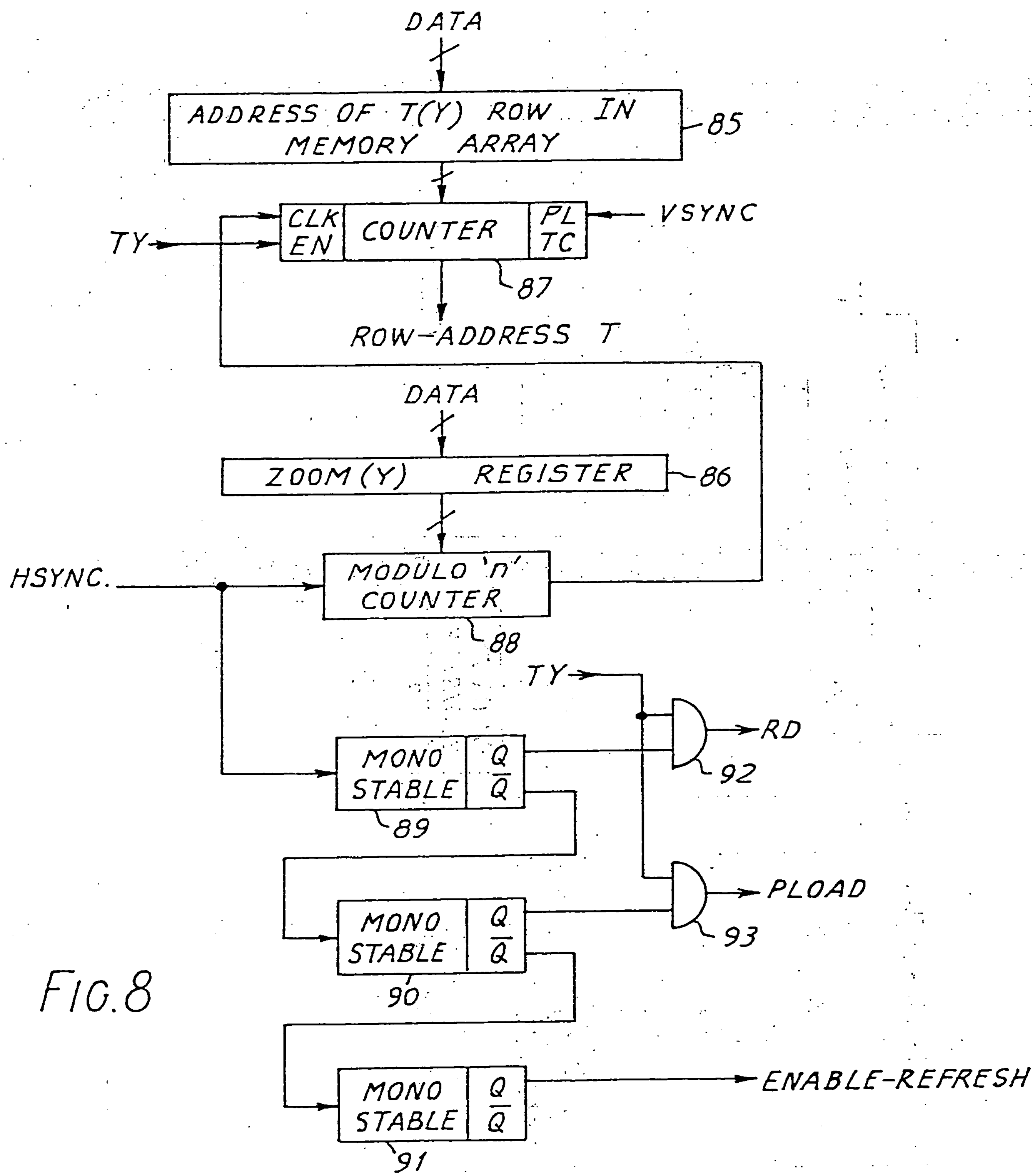

DATA
ADDRESS OF T(Y) ROW IN MEMORY ARRAY
85
CLK EN
COUNTER
PL TC
VSYNC
TY
87
ROW-ADDRESS T
DATA
ZOOM (Y) REGISTER
86
HSYNC.
MODULO 'n' COUNTER
88
TY
MONO STABLE
Q
Q̄
89
RD
92
MONO STABLE
90
PLOAD
93
MONO STABLE
ENABLE-REFRESH
91

FIG.8

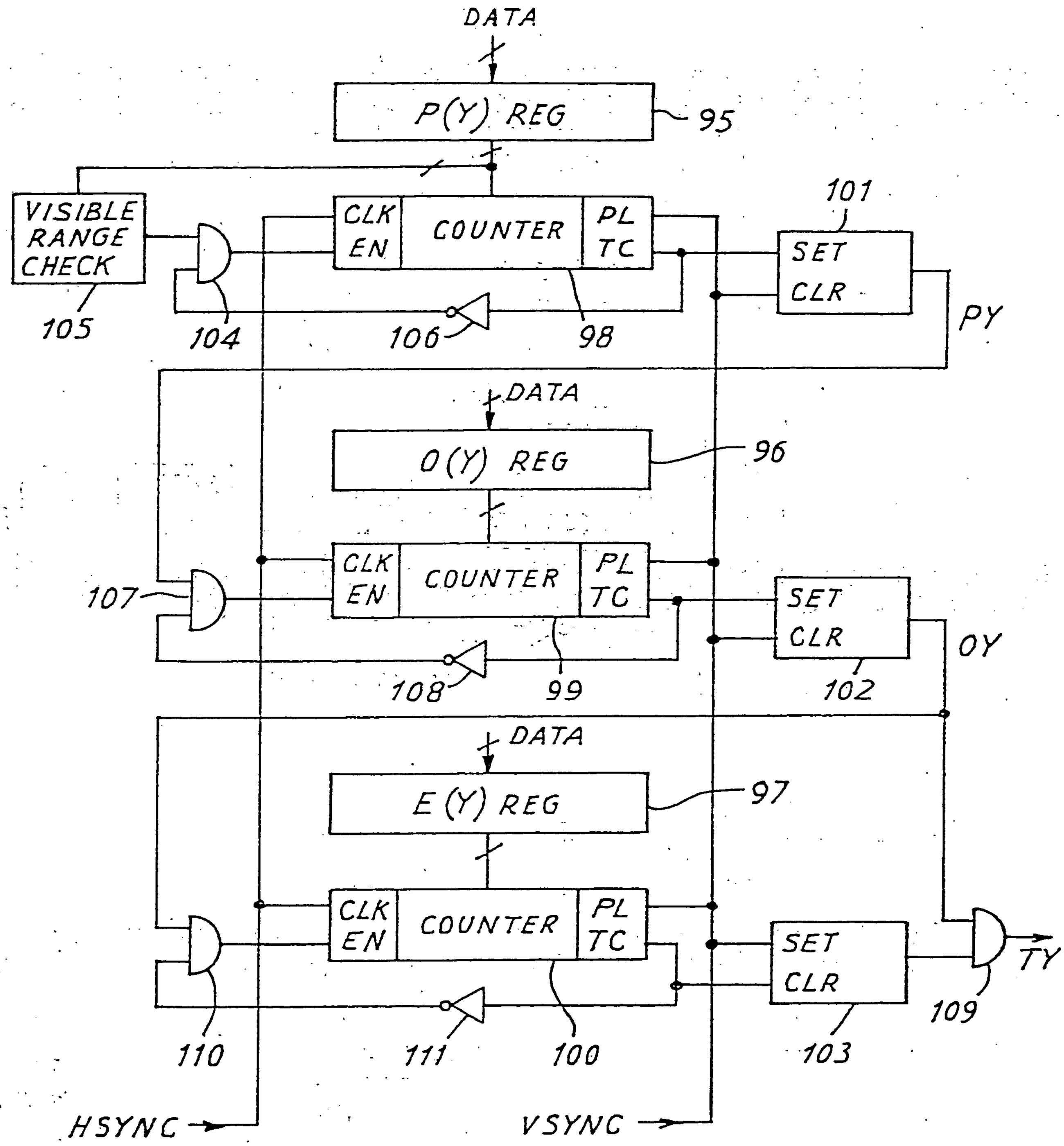

DATA
P(Y) REG
95
VISIBLE RANGE CHECK
CLK EN
COUNTER
PL TC
101
SET CLR
PY
105
104
106
98
DATA
O(Y) REG
96
107
CLK EN
COUNTER
PL TC
SET CLR
OY
108
99
102
DATA
E(Y) REG
97
CLK EN
COUNTER
PL TC
SET CLR
TY
110
111
100
103
109
HSYNC
VSYNC

FIG.9

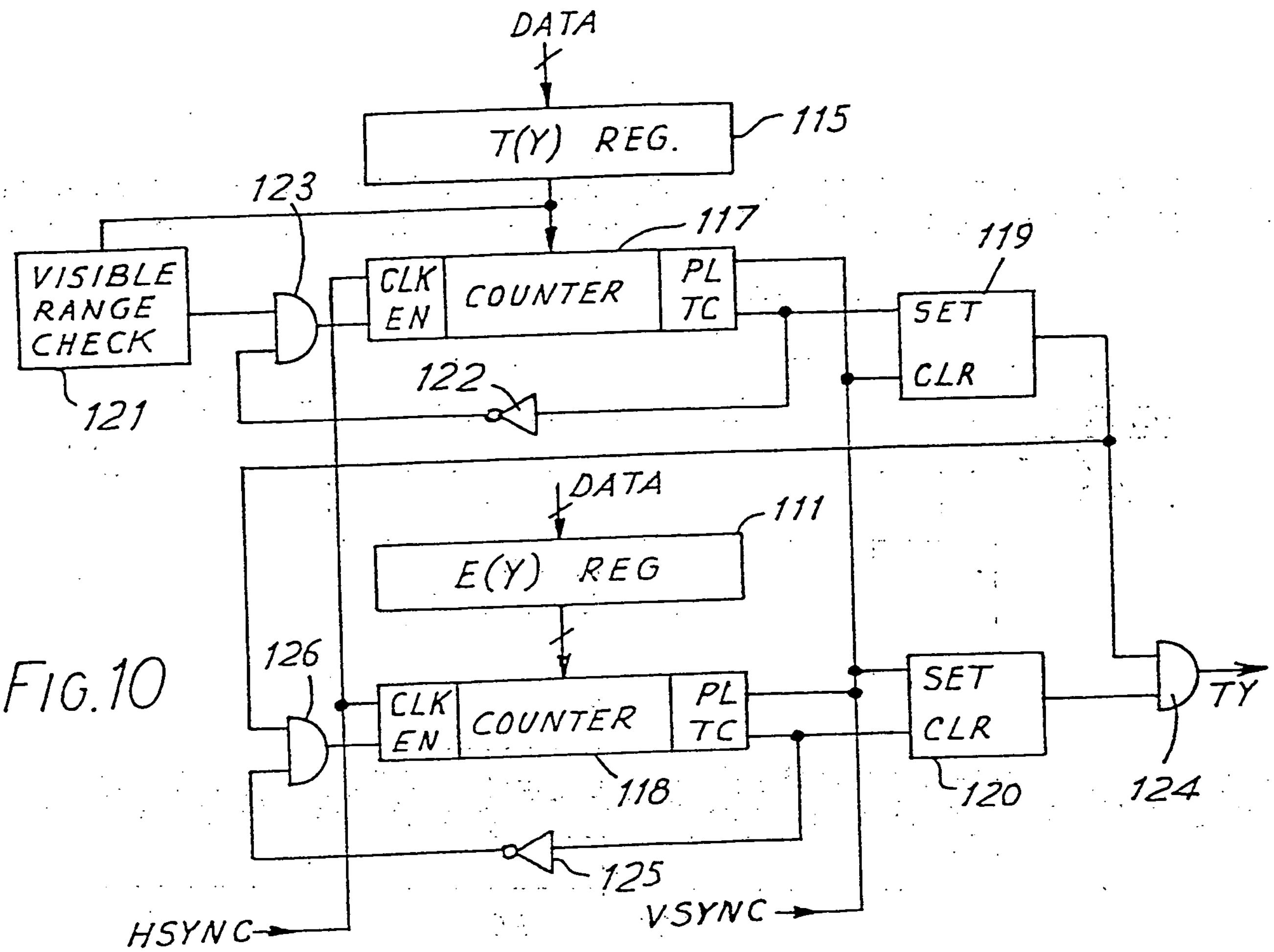


FIG.10

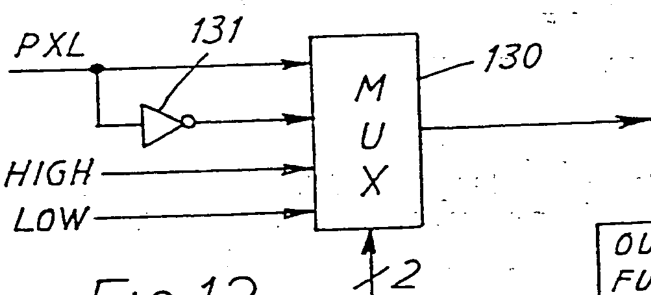
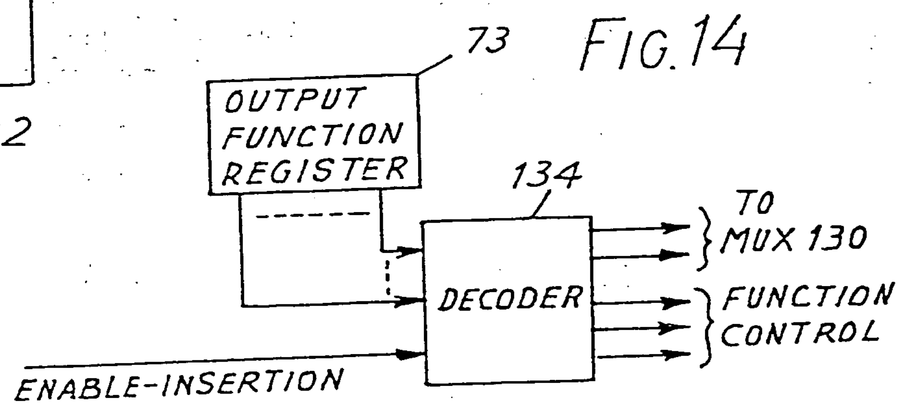


FIG.12

FIG.14



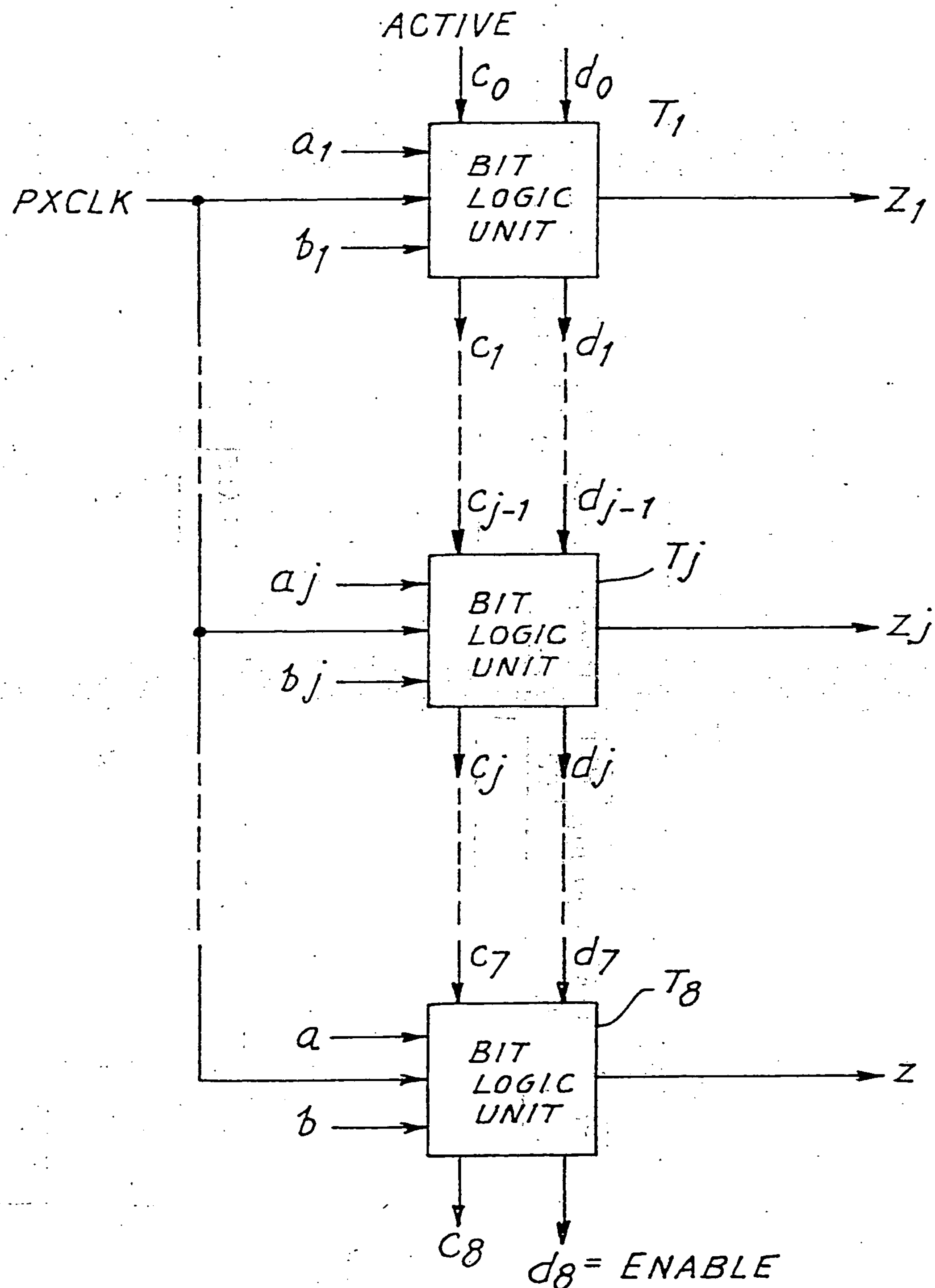

ACTIVE
c_0
d_0
T_1
a_1
PXCLK
b_1
BIT LOGIC UNIT
Z_1
c_1
d_1
c_{j-1}
d_{j-1}
T_j
a_j
b_j
BIT LOGIC UNIT
Z_j
c_j
d_j
c_7
d_7
T_8
a
b
BIT LOGIC UNIT
z
c_8
d_8 = ENABLE

FIG.11

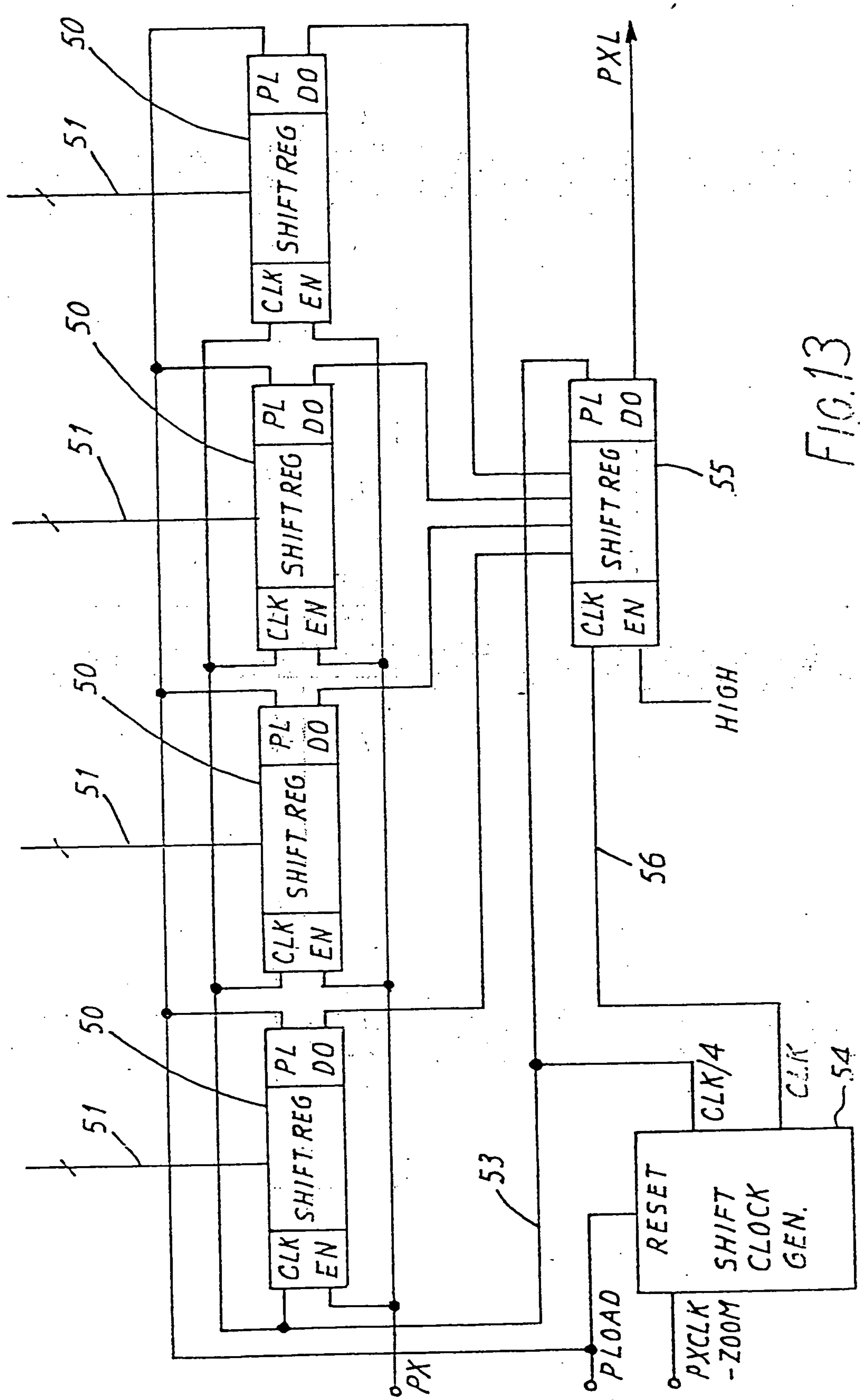

50
51
PL
DO
SHIFT REG
CLK
EN
PXL
55
HIGH
56
53
CLK/4
CLK
54
RESET
SHIFT
CLOCK
GEN.
PX
PLOAD
PXCLK
-ZOOM

FIG.13

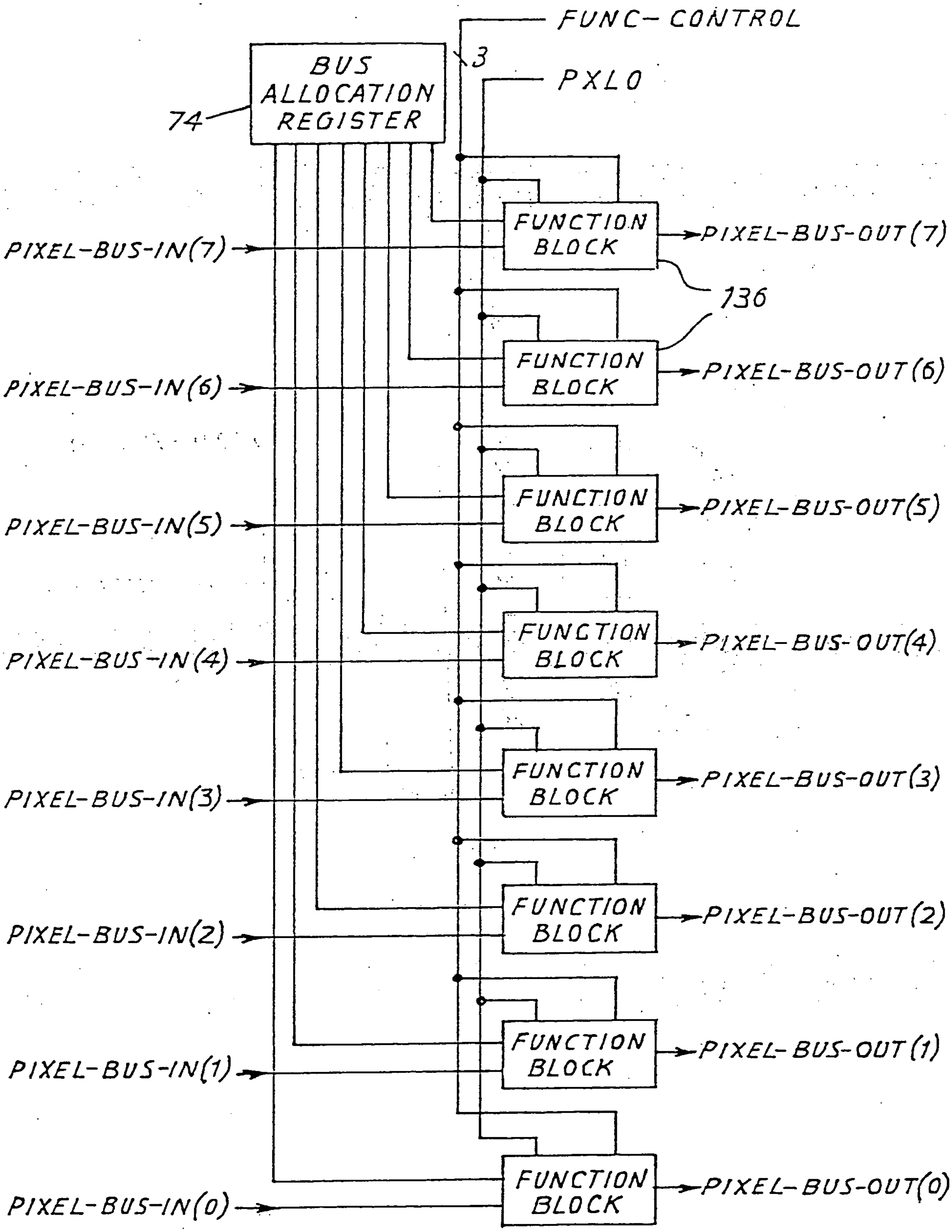

FUNC- CONTROL
3
PXLO
BUS ALLOCATION REGISTER
74
PIXEL-BUS-IN(7)
FUNCTION BLOCK
PIXEL-BUS-OUT(7)
136
PIXEL-BUS-IN(6)
FUNCTION BLOCK
PIXEL-BUS-OUT(6)
PIXEL-BUS-IN(5)
FUNCTION BLOCK
PIXEL-BUS-OUT(5)
PIXEL-BUS-IN(4)
FUNCTION BLOCK
PIXEL-BUS-OUT(4)
PIXEL-BUS-IN(3)
FUNCTION BLOCK
PIXEL-BUS-OUT(3)
PIXEL-BUS-IN(2)
FUNCTION BLOCK
PIXEL-BUS-OUT(2)
PIXEL-BUS-IN(1)
FUNCTION BLOCK
PIXEL-BUS-OUT(1)
PIXEL-BUS-IN(0)
FUNCTION BLOCK
PIXEL-BUS-OUT(0)

FIG.15

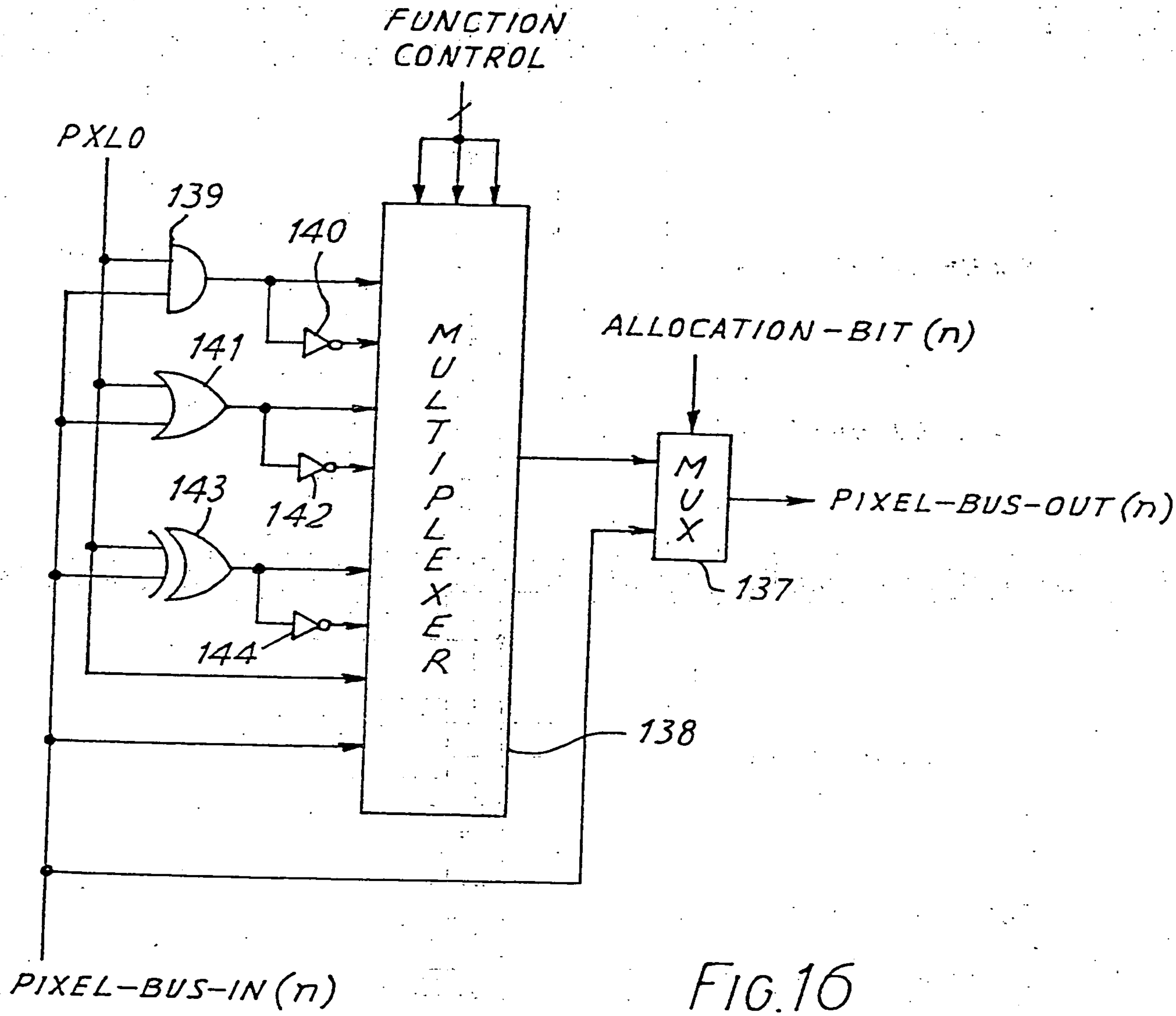

FUNCTION CONTROL
PXLO
139
140
141
142
143
144
M U L T I P L E X E R
ALLOCATION-BIT (n)
MUX
PIXEL-BUS-OUT (n)
137
138
PIXEL-BUS-IN (n)

FIG.16

SPECIFICATION

Window graphics system

The present invention relates broadly to window graphics systems, that is to say systems enabling independently generated image data pertaining to a plurality of windows in "world space" to be displayed in a plurality of viewports, which are rectangular areas of a display. It is well known to provide images on a rasterscan display screen from a frame store, also referred to as a frame buffer, which stores one or more bits per pixel. Windows may be software defined by appropriate manipulation of the data written into a frame store but this requires extensive memory operations, in particular bit-block transfer operations to perform scrolling and other window movement functions. This is either time-consuming if performed by a general-purpose processor, or expensive if performed by a high-performance bit-block transfer processor.

Various techniques are known to assist in updating image data, dealing with hidden surface removal in representing objects which partially obscure each other and superimposing a small image (especially a cursor) in a variable position. These include use of multiple plane frame stores, use of video mixing under different logical functions to overlay opaque and transparent images and provision of registers to indicate the coordinate position of an image plane relative to the screen coordinate system. This latter feature provides a transformation between "world space" and "screen space" defined by the coordinate systems of a frame store and the screen respectively. In particular a window in world space may be transformed to a viewport in screen space.

It is known to provide stores for areas less than full screen size, e.g. a cursor store.

The object of the present invention is to provide a frame store assembly for a raster-scan display which makes it possible to achieve very flexible, high-speed window graphics with relatively simple control means.

According to the present invention there is provided a frame store assembly for a raster-scan display comprising a plurality of like bit-plane memory devices, each capable of storing pixel image date for a part area of the display, means for writing image data independently into the memory devices, and a pixel bus routed through each memory devices, each such device comprising programmable control means operable synchronously with respect to synchronising signals for the raster-scan display to read out on to the pixel bus selected data from the memory device at times which are determined by the programming of the memory device and thus cause the read-out data to provide video data for a selected part area of the display, whereby the display is synthesized from the read-out data from selected memory devices.

It will be convenient to adopt the following terminology. Each said memory device is a micro-frame store, abbreviated herein as MF store. Each MF store stores one "pane" in world space. The control means of a plurality of MF stores may be programmed in such a way that the panes fit together to form a window. In screen space a viewport is formed from one or more "tiles". A pane contains image data for one tile only; a tile can display all, part or none of the data in a pane. A plurality of MF stores may be assigned to the same pane to provide a plurality of bits per pixel.

Since the contribution of an MF store to the display depends upon the way in which its control means is programmed it becomes possible to define whatever windows are required and to assign as many MF stores to a pane as are required to provide the desired number of bits per pixel (up to a given maximum). Only those MF stores which are needed for a given application need be used. As will be explained below, it becomes very simple to deal with the problem of overlapping windows.

The embodiment of the invention to be described below has the following features and advantages:

The invention will be described in more detail, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a diagram of a window and a viewport,
Figure 2 is a diagram showing the relationship between one pane of a window and one tile of a viewport,
Figure 3 is a block diagram of one embodiment of the invention,
Figure 4 is a more detailed block diagram of one MF store,
Figure 5 is a schematic block diagram of an embodiment of a MF store memory array,
Figure 6 is a block diagram of a scanline section of a video output unit,
Figure 7 is a block diagram of a pixel-rate section of the video output unit,
Figure 8 is a block diagram of a first signal generating circuit of the scanline section of Figure 6,
Figure 9 is a block diagram of a second signal generating circuit,
Figure 10 is a block diagram of an alternative second signal generating circuit,
Figure 11 shows a priority resolution unit of Figure 7 in more detail,
Figure 12 shows a normal/inverse video circuit of Figure 7.
Figure 13 shows part of Figure 5 in more detail,
Figure 14 shows an output function control circuit of Figure 7 in more detail,
Figure 15 is a block diagram of pixel combination logic included in Figure 7, and
Figure 16 shows part of Figure 15 in more detail.

Figure 1 shows a window W composed of twelve panes P(0,0) to P(3,2), disposed within the area of a screen S at a location defined by the coordinates x_0, y_0 of the top left-hand corner of the viewport. Although screen graphics are often related, so far as user software is concerned, to a bottom left origin, it is convenient in implementing the present invention to work with a top left origin, given the normal scanning pattern of a

raster-scan display. The positions of the individual panes can be specified by their individual origin coordinates x_0,y_0, x_1,y_0, x_0,y_1 and so on.

A viewport V (shaded) is shown superimposed on the window W. The viewport consists of the whole of the pane P(2,1) and parts of the eight surrounding panes. The viewport is divided into nine tiles separated by the parts of the partitions between tiles drawn in heavy lines. The viewport is displayed by reading out selected contents of the MF stores corresponding to the relevant panes P(1,0), P(2,0), P(3,0), P(1,1) etc at the correct times relative to the synchronizing signals which define the screen origin (0,0).

Figure 2 illustrates the most general case which can arise when a viewport of one tile T is set wholly inside a window of one pane P. The origin of the tile is shown as T(x,y) and the origin of the pane is shown as P(x,y), both in screen coordinates. T(x,y) is offset from P(x,y) by two quantities, the pixel offset 0(x) and the scanline offset 0(y). The size of the tile is specified by two quantities, the pixel extent E(x) and the scanline extent E(y).

In terms of the time bases of the raster-scan display, times t_1 to t_4 in each line scan correspond to left and right edges of the pane and tile while scanning lines 1_1 to 1_4 correspond to top and bottom edges of the pane and tile. For simplicity the well-known considerations which arise when interlaced scanning is used are not discussed here.

The problem of extracting the required data from the MF store (or stores) holding the pane P can now be understood as follows. The counters which read out the contents of the MF store have to be synchronized so as to move on during t_1 to t_4 in each of lines 1_1 to 1_4. The output bit stream has to be multiplexed into the output video stream only during t_2 to t_3 in each of lines 1_2 to 1_3. The means for achieving this will be described below.

It will be noted that there is redundancy in the information shown in Figure 2. The control means of a MF store can be programmed with P(x, y) and T(x,y) or with either of these coordinate pairs plus 0(x) and 0(y). If P(x,y) is not needed for its own sake it is possible to use T(x,y) direct or P(x,y) in conjunction with 0(x) and 0(y). It is also necessary to program E(x) and E(y).

The screen may have a size of the order of 1000 x 1000 pixels or 2000 x 2000 pixels. Convenient sizes for the MF stores may then be 16k bits (128 x 128 bit panes) or 64k bits 256 x 256 bit panes). To create a full screen 8 x 8 = 64 MF stores may be required for one bit per pixel. To provide a plurality of bits per pixel more stores are required, of the order of hundreds of MF stores for a high pixel resolution, high colour resolution system. For example up to eight bits per pixel may be required for a system which utilizes a colour look-up table to transform 8-bit parallel video data into R, G and B colour values. For even greater colour resolution eight bits per pixel may be required for each of R, G and B, enabling the intensity of each colour to be specified independently for each pixel and to an 8-bit resolution.

The MF stores may define rectangular panes with the same aspect ratio as the screen (as implied by Figures 1 and 2), but it may be more convenient to utilize stores which define square panes, as implied by the numerical examples given above.

Figure 3 is an overall block diagram of an embodiment of the invention. A plurality of MF stores 10 are disposed in a linear array along various buses and lines to be described below. A host graphics processor 11 communicates with the MF stores 10 by way of a store controller 12. Specifically, commands (including parameters) are sent on a command line 13. Frame and line synchronizing pulses are sent on a line 14. These reach the most remote MF store 10Z and return to the first store 10A on a line 15 for reasons which will become apparent below. Image data to be written into MF stores is sent out on a line bus 16. Image data can also be read out of a store back to the host on a line or bus 17. By means of the command line 13 and data-in line 16 it is possible to set up each MF store to produce its own video signals for possible insertion on to a pixel bus 18. Whether insertion takes place or not is determined in part by priority contests. A priority bus 19 carries a priority number from MF store to MF store. The buses 18 and 19 run from the most remote store 10Z to the first store 10A and video generation is synchronous with the sync signals on the returning syncs out line 15.

Figure 4 shows the main sections of one MF store 10, namely a communication interface 20, a local control unit 21, a memory array 22 and a video output unit 23. No detailed description will be given of the communication interface 20, nor of the processor 11 or controller 12 of Figure 3. The present invention is not concerned with the high level processing of graphics commands, nor with the way in which commands are passed to the MF stores. This may be implemented in ways well known in themselves in computer bus systems. It is sufficient for an understanding of the present invention to realize that the computer can write values into any of the registers shown in the more detailed drawings, addressing such values to the registers individually. Moreover use may be made of well known facilities for sending global commands and group commands which are obeyed by a particular group of devices. An example of a complete repertoire of control operations will be given below. Firstly, however, the basic elements of the system which enable video data to be output on to the pixel bus will be described.

The command line 13 and data lines 16 and 17 may be clocked through one bit buffers in the communication interfaces 20. In each such interface, decoding circuits are tapped off the command line for affecting address comparisons and command decoding. Commands may thus be pipelined down the line 13.

The memory array 22 is essentially a conventional bit-plane video RAM into which image data can be written via the local control unit. The video output unit 23 accepts selected data from selected rows of the memory array 22 and serializes such data for possible insertion on to the pixel bus 18. The video output unit comprises a section operating at scanline rate and a section operating at pixel rate, and will be fully described below. Firstly however, a brief description of the memory array 22 will be given for completeness.

Figure 5 is a schematic block diagram of the memory array 22 of one MF store 10. The array is a 4 by 4 array

of RAM cells 45, each of which may have a capacity of 1k bits for 128 × 128 bit panes or a capacity of 4k bits for 256 × 256 bit panes. Assuming the former alternative, each cell 45 is a 32-row by 32-column memory array requiring a 5-bit row address and a 5-bit column address. Three registers are accessible to the graphics store controller, namely a 16-bit data register 46 to which data may be sent on a DATA IN 16 and from which data may be read a DATA OUT 17, an address register 47 to which a 10-bit address may be sent and a 16-bit enable register 48. The commands and parameters line 13 enables the registers to be loaded as required and also enables the communications interface 20 and local control unit 21 to provide READ and WRITE signals. The read and write operations take place as follows. For simplicity the relevant connections are shown only to the top left-hand corner RAM cell 45.

The REM cells 45 are addressed in parallel by the address register 47. The bits of the data register 46 are in one-to-one correspondence with the RAM cells 45 and, when WRITE is asserted, the bits in the data register are written via data in DIN lines into the addressed locations of the sixteen cells. This is subject to the cells being enabled. The bits of the enable register 48 are also in one-to-one correspondence with the RAM cells 45 and provide enable signals EN to the cells. The enable register 48 can be set up as a kind of mask which enables only a block of the cells 45. In line drawing the data in the enable register indicates those pixels which lie on the path of the line. The write cycle then writes data only to the enabled cells.

When READ is asserted, the contents of the addressed locations of the enabled RAM cells are transferred on data out DOUT lines to the corresponding bit positions in the data register 46.

The features of the memory array 22 so far described relate to its communications with the MF store controller 12 and hence the graphics processor 11. Means are provided, in a manner known per se, for independent read-out of data to the video output unit 23. These comprise four 32-bit shift registers 50 into which a complete row of data from corresponding columns of the RAM cells 45 may be parallel-written over highways 51. The RAM cells 45 may be dynamic RAM refreshed in conventional manner, with avoidance of conflicts between refresh and data transfers over the highways 51. These transfers take place under control of a PLOAD signal, a RD ROW (read row) signal and ROW ADDRESS, which are generated in a manner described below. RD ROW in conjunction with the ROW ADDRESS read the selected row on to the highways 51 and PLOAD parallel-loads the row into the shift registers 50.

The shift registers 50 are clocked in parallel by pulses on a line 53, namely quarter pixel rate pulses PXCLK/4, when the registers are enabled by PX. As will be described below, PX defines the time in each scanline corresponding to the horizontal dimension of the pane. The bits clocked out of the shift registers 50 are parallel loaded into a 4-bit shift register 55 which is clocked by pixel rate pulses PXCLK on a line 56. The output from the shift register 55 is pixel rate video data PXL which is returned to the video output unit 23. This part of the memory array 22 is shown more fully in Figure 13, described below.

The memory array 22 does not have to be a square array. Another possible arrangement is a linear array comprising only one set of four RAM cells 45. However this defines a horizontally-long, vertically narrow pane. A square or near square rectangular pane has the advantage that fast vertical, as well as horizontal, line-drawing will be possible. There may be redundant cells 45 and means for allocating a spare cell in place of a defective cell, a technique well known in LSI memory devices.

PXL consists, in each frame, of 128 bursts of 128 pixel pulses each occurring at the correct times in 128 consecutive scanlines to define the pane P, assuming the specific embodiment of Figure 6 and a non-interlaced raster. If interlaced scanning is employed there will be 64 odd-line bursts in odd fields and 64 even-line bursts in even fields.

The bursts of pixel pulses have to be appropriately gated to select only those pulses which pertain to the tile T (Figure 2). The pulses may also be selectively inverted provide normal or inverse video and provision may be made for replication of pixels independently in the horizontal and vertical directions, as determined by zoom factors. Expansion of an image by this technique is well known. In the case of 2 × 2 zoom, for example, each pixel is duplicated in each line to expand each burst of 128 pulses into a burst of 256 pulses and each burst is moreover repeated on two consecutive scanlines.

The video output unit 23 which provides these and other facilities will now be described. It consists of a scanline section (Figure 6) and a pixel-rate section (Figure 7). The scanline section comprises a set of registers 60 into which the following values can be written (via the local control unit 21):

PANE (Y)	The scanline number T_1 (Figure 2)
OFFSET (Y)	O(y) in Figure 2
EXTENT (Y)	E(y) in Figure 2
ZOOM (Y)	The vertical zoom factor

A vertical pane control unit 61 is responsive to the values in these registers and to the horizontal and vertical synchronizing pulses HSYNC and VSYNC to provide the following signals:

ROW ADDRESS T	The address in the memory array 22 of the curent row within the tile T (Figure 2)
RD	A read signal accompanying ROW ADDRESS T
PLOAD	A signal commanding loading of a row from the memory array 22 into the pixel-rate section
TY	A signal which remains true during the lines 1_2 to $(1_3\text{-}1)$ which are involved in the title T
ENABLE REFRESH	A strobe which enables a conventional refresh control circuit 62

The refresh circuit 62 provides the following signals:

ROW ADDRESS R	The current row refresh address
RF RD	A read signal accompanying ROW ADDRESS R
R SEL	A selection signal.

The selection signal R SEL controls two multiplexers 63 and 64 which, when a refresh operation is taking place, route ROW ADDRESS T to ROW ADDRESS and route RF RD to RD ROW, these output signals being input to the memory array 22 as already described with reference to Figure 5. When no refresh operation is taking place, the multiplexers 63 and 64 route ROW ADDRESS T to ROW ADDRESS and route RD to RD ROW.

The pixel-rate section is shown in Figure 7. A circuit 70 MICRO-FRAME STORE DATA OUTPUT represents the registers 50 and 55 of Figure 5, described more fully with reference to Figure 13 below. The pixel rate section includes various registers 71 to 75 which are loaded with the following quantities via the local control unit 21:

PANE (X)	The number corresponding to t_1 (Figure 2)
OFFSET (X)	0(x) in Figure 2
EXTENT (X)	E(x) in Figure 2
ZOOM (X)	The horizontal zoom factor
OUTPUT FUNCTION	Selects various modes of handling PXL
BUS ALLOCATION	Determines which pixel lines in the pixel bus 18 are affected by PXL
PRIORITY	Determines the priority of the MF store relative to the other MF stores.

The registers 71, i.e. the first three registers listed above control a horizontal pane control circuit 76 which also receives TY and outputs PX and TX where PX extends from t_1 to t_4 (Figure 2) and TX extends from t_2 to t_3, so as to mark off the pane and tile respectively within a scanning line.

PX enables data to be clocked out from the circuit 70 as PXL in response to a clock PXCLK ZOOM. A zoom control circuit 77 is a controlled divider which divides PXCLK, the pixel rate clock, by the zoom factor to produce PXCLK ZOOM. The zoom factor will commonly be unity, but when horizontal enlargement by a factor of 2, 3 ..., the zoom factor is set accordingly, so that PXCLK ZOOM is half the rate, a third the rate, and so on, of PXCLK. Thus, reverting to Figure 5, the signals on lines 53 and 56 are only PXCLK/4 and PXCLK when the horizontal zoom factor is unity. In the more general case the inputs are:

(PXCLKZOOM)/4 = PXCLK/4m (line 53)
and PXCLK ZOOM = PXCLK/m (line 56)
where m is the horizontal zoom factor.

PXL is passed through a circuit 78 which selectively inverts PXL to provide PXLO and can also clamp PXLO high or low. The function performed by the circuit 78 is determined by an output function control circuit 79 in accordance with the contents of the OUTPUT FUNCTION register 73. PXLO is fed to pixel combination logic 80 for insertion on to the pixel bus 18 subject to three sets of conditions:

(1) PXLO affects only those pixel lines selected by the BUS ALLOCATION register 74.

(2) The mode of insertion can be overwrite, logical AND, logical OR, exclusive OR or their inverses. The mode is also selected by the output function register 73 and control unit 79.

(3) Insertion only takes place during a signal ENABLE INSERTION provided by a priority resolution unit 81 which compares the number on the priority bus 19 with the priority number in the register 75 during the interval marked by TX.

The vertical pane control 61 of Figure 6 will now be described more fully with reference to Figures 8 to 10. Two registers 85 and 86 are loaded from the local control unit 21 with the following two parameters respectively:

T(Y)ADDRESS	The address in the memory array 22 of the first row within the tile T
ZOOM(Y)	The vertical zoom factor.

The T(Y)ADDRESS is loaded into a counter 87 at the beginning of each frame by VSYNC. During TY the counter 87 counts HSYNC, or a sub-multiple of HSYNC as determined by ZOOM (Y), and the parallel output of the counter is ROW ADDRESS T.

The ZOOM(Y) register 86 controls a programmable counter 88 so as to select the division factor thereof in accordance with the vertical zoom factor n, which may be 1, 2, 3 etc. It follows that the same row address is

used n times.

HSYNC is also applied to a chain of monostable circuits 89, 90, 91 which respectively provide RD, PLOAD and ENABLE REFRESH in rapid succession. However RD and PLOAD are gated with TY in AND gates 92 and 93 so as to be output only during the vertical extent of the tile T.

Figure 9 shows one embodiment of the part of the vertical pane control 61 which generates TY, namely for the case in which P(y) is used in conjunction with O(y) to determine T(y) (Figure 2). The circuit comprises registers 95, 96 and 97 into which are written, via the local control unit 21, the following values respectively:

P(Y)
O(Y)
E(Y)

These quantities are loaded into corresponding counters 98, 99 and 100 at the beginning of each frame by VSYNC which also clears a PY latch 101, clears an OY latch 102 and sets a TY enable latch 103. The counters are all clocked by HSYNC but are selectively enabled. Consider the counter 98 to be enabled. It counts scanning lines (HSYNC) and the value of P(Y) is such that the counter reaches its terminal count at line 1_1 (Figure 2), whereupon the latch 101 is set to provide PY. The enabling signal for the counter 98 is provided via an AND gate 104 by visible range check circuit 105 which performs simple range checking on P(Y) to determine whether the pane P (Figure 2) is at least partially within the visible screen area, so far as vertical coordinates are concerned. If not, the counter 98 is never enabled and TY is never emitted. When the counter 98 reaches its terminal count and sets the latch 101, it also disables itself via an inverter 106 and the AND gate 104.

PY provided by the latch 101 enables the O(Y) counter 99 via an AND gate 107. This counter reaches its terminal count at line 1_2 (Figure 2) and thereupon sets the latch 102 and disables itself via an inverter 108 and the AND gate 107. The signal OY provided by the latch 102 then provides TY by way of an AND gate 109 which is enabled by the set latch 103. OY also enables the counter 100 via an AND gate 110. When this counter reaches its terminal count it disables itself via an inverter 111 and the AND gate 110 and also clears the latch 103. The AND gate 109 is therefore disabled and TY is terminated.

It will be appreciated that the quantities P(Y) and O(Y) loaded into the registers 95 and 96 must be complementary to the quantities P(y) and O(y) of Figure 2 if the counters counts upwardly.

As already noted, use may be made of T(y) rather than P(y) and O(y). The alternative circuit of Figure 10 is then employed. T(Y) and E(Y) are entered in registers 115 and 116 and loaded into corresponding counters 117 and 118 by VSYNC. Latches 119 and 120 are cleared and set respectively by VSYNC. So long as it is enabled by a visible range check circuit 121, the counter 117 counts HSYNC until it reaches terminal count when it disables itself via inverter 122 and AND gate 123 and sets the latch 119. TY is then provided by an AND gate 124 enabled by the set latch 120.

The latch 119 enables the counter 118. When this counter reaches its terminal count it disables itself via an inverter 125 and an AND gate 126 and clears the latch 120 to disable the AND gate 124 and terminate TY.

The horizontal pane control circuit 76 of Figure 7 will not be described in detail since it is essentially the same as the circuit of Figure 9 but with P(Y), O(Y) and E(Y) replaced by P(X), O(X) and E(X), and VSYNC and HSYNC replaced by HSYNC and PXCLK respectively. Moreover the AND gate 104 has to have a third input, namely TY. PX and TX are taken from the latch 101 and the gate 109 respectively.

It is assumed, by way of example, that the priority bus 19 is eight bits wide. This bus inputs a bus priority number A to the priority resolution unit 81 which receives the MF store priority number B from the register 75. So long as TX is false, the MF store priority number B is ignored, ENABLE INSERTION is forced false and the input bus priority number A is passed on to the next priority resolution unit as the output bus priority number Z. If TX is true, a priority contest takes place, ENABLE INSERTION is true if the MF store priority number B has higher priority than the bus priority number A, otherwise ENABLE INSERTION is false, and Z is output as whichever of A and B has the higher priority. Higher priority may correspond to a larger number (maximum priority resolution) or to a smaller number (minimum priority resolution).

An example of the priority resolution unit 81 is shown in Figure 11. The bits of the numbers A, B, and Z will be denoted as follows:

$$A = a_1 \ \ldots \ a_j \ \ldots \ a_8$$
$$B = b_1 \ \ldots \ b_j \ \ldots \ b_8$$
$$Z = z_1 \ \ldots \ z_j \ \ldots \ z_8$$

where the subscripts 1 and *8* denote the most and least significant bits respectively and *j* denotes a general bit.

The priority resolution unit 80 comprises eight bit logic units T_1 to T_8. Considering the general unit T_j, it receives the bits a_j and b_j and outputs z_j. The unit T_j also receives inputs c_{j-1} and d_{j-1} from the next more significant unit T_{j-1} and outputs c_j and d_j to the next less significant unit. The signal c_j indicates whether or not priority has been resolved at the unit T_j or at a unit of higher significance and d_j indicates, when priority has been resolved, in whose favour. The final bit d_8 constitutes the ENABLE INSERTION signal. z_j becomes a_j for the priority resolution unit of the next MF store.

One set of equations defining the structure for T_j in the case of minimum priority resolution is as follows:

$$c_j = c_{j-1} + (a_j \oplus b_j) \quad (1)$$
$$d_j = c_{j-1}.d_{j-1} + a_j.\overline{c_{j-1}} \quad (2)$$
$$z_j = a_j.b_j + c_{j-1}(b_j.d_{j-1} + a_j.\overline{d_{j-1}}) \quad (3)$$

The convention assumed is that $c_{j-1} = 0$ means "not resolved" while $c_{j-1} = 1$ means "resolved". If $c_{j-1} = 0$ the value of d_{j-1} is without significance. If $c_{j-1} = 1$, then $d_{j-1} = 1$ means the device has won the priority contest whereas $d_{j-1} = 0$ means the bus has won the priority contest.

The provision of the initial values c_o and d_o will now be considered. c_o is the signal TX while d_o is tied to logical 0. This causes ENABLE INSERTION to be forced false by TX false, whereas during TX true, ENABLE INSERTION is true only if the MF store priority number B has higher priority than the bus priority number A, (save that the state of ENABLE INSERTION is delayed eight bits relative to the state of TX).

A fuller explanation of the priority resolution unit, and an example which resolves in favour of the maximum priority number, will be found in our copending application 8518130, entitled "Priority Resolution System".

The priority resolution unit as described with reference to Figure 11 is a pipelined unit operating with the most significant bus priority bit a_1 eight bits early relative to the pixel at which the priority resolution obtains. A non-pipelined unit could equally well be used, in which the *c* and *d* signals would ripple through the bit logic units T_1 to T_8.

Figure 12 shows the circuit 78 of Figure 7. A multiplexer 130 is controlled by two bits from the output function control circuit 79 to select as PXLO either PXL, PXL inverted by an inverter 131, HIGH or LOW.

Figure 13 shows the micro-frame store data output circuit 70 in more detail. The shift registers 50 and 55 have parallel-load terminals PL, shift enable inputs EN, shift clock inputs CLK and data outputs DO. The shift registers 50 are parallel-loaded from the highways 51 in response to PLOAD.

PXCLK ZOOM from the zoom control circuit 77 is fed to a shift clock generator 54 which is essentially a two-bit (divide-by-four) counter producing an output CLK (at the same rate as PXCLK ZOOM) on the line 56 and a quarter-rate output CLK/4 on the line 53. The counter is reset by PLOAD to establish the phase of CLK/4 correctly.

PX enables the registers 50 to clock out their contents in response to CLK/4 which parallel-loads each group of four bits from DO of the registers 50 into the register 55. This register is permanently enabled and clocked by CLK to feed out PXL.

Figure 14 is a diagram of the output function control circuit 79. This consists of a decoder 134, e.g. a PROM, which is addressed by the contents of the output function register 73 and ENABLE INSERTION from the priority resolution circuit 81. The decoder provides two output bits controlling the multiplexer 130 (Figure 12) in the normal/inverse circuit 78. The decoder 134 furthermore provides three bits FUNCTION CONTROL controlling the pixel combination logic 80 which is shown in Figure 15.

Figure 15 is drawn for the case of an 8-bit wide pixel bus 18. The bus comprises PIXEL BUS IN (0) to (7) on the input side and PIXEL BUS OUT (0) to (7) on the output side. For each bit, a function block 136 receives PIXEL BUS IN (n) and PXLO and provides PIXEL BUS OUT (n) in a manner determined in part by the function control signal from the output function control circuit 79 and in part by a corresponding bit of the bus allocation register 74. If this bit is zero, the function block 136 simply passes PIXEL BUS IN (n) to PIXEL BUS OUT (n) but, if the bit is one, PXLO can influence PIXEL BUS OUT (n) in a manner determined by the function control signal. Figure 16 shows one embodiment of the function block 136.

A multiplexer 137 provides PIXEL BUS OUT (n) and is controlled by the allocation bit (n) from the register 74. When this bit is zero, the multiplexer selects PIXEL BUS IN (n) but, when the bit is one, it selects the output of an 8-to-1 multiplexer 138 which is controlled by the 3-bit function control signal applied to all function blocks 136 in parallel from the output function control circuit 79. The eight inputs to the multiplexer 138 are as follows:

(1) PIXEL BUS IN (n)
(2) PXLO
(3) The AND of PXLO and PIXEL BUS In (n) provided by an AND gate 139
(4) The inverse of (3) provided by an inverter 140
(5) The OR of PXLO and PIXEL BUSI N (n) provided by an OR gate 141
(6) The inverse of (5) provided by an inverter 142
(7) The exclusive OR of PXLO and PIXEL BUS IN (n) provided by an exclusive OR gate 143
(8) The inverse of (7) provided by an inverter 144

The decoder 134 (Figure 14) always selects input (1) of the multiplexer 138 when ENABLE INSERTION is false, so that PIXEL BUS OUT (n) then equals PIXEL BUS IN (n).

An alternative to Figures 14 and 16 is to dispense with the decoder 134 and to enter fully-decoded control signals in the output function register 73 for providing the signals to control the multiplexer 130 in Figure 12 and the multiplexer 138 in Figure 16. The ENABLE INSERTION signal then acts directly in the pixel combination logic 80. For example in Figure 16, the multiplexer 137 can be controlled not by allocation bit (n) itself, but by the AND of this bit with ENABLE INSERTION.

Although the asemblage of MF stores 10 may be the sole means of putting video data on to the pixel bus 18 it is also possible to supplement the MF stores with one or more full frame stores, which may be assigned a minimum priority value. Other video data may of course be put on the pixel bus for combination with MF store

data.

It is a simple matter with the described system to provide a variety of scrolling and such operations. A viewport and the image displayed therein may be moved together over the screen, a process known as dragging. This is achieved by altering the P(X) and/or P(Y) coordinates of the or each pane contributing to the viewport. It is possible to move a viewport over the image data, which does not move, a process known as roaming. This is achieved by altering the O(X) and/or O(Y) coordinate value. It is possible to scroll or pan (sideways scrolling) the image data through a fixed viewport. This is achieved by altering P(X) and/or P(Y) with complementary changes of O(X) and O(Y). Any one of these changes may be set up as a motion vector whereby the relevant coordinate or coordinates is changed by a fixed amount per frame.

In order to facilitate the treatment of viewports composed of more than one tile, it is desirable to be able to associate microframe stores pertaining to the same window, in such a way that group addressing may be employed for these stores.

When scrolling an image through a viewport, whether by vertical scrolling or sideways scrolling, continuous scrolling may be achieved by providing one more MF store than is needed to cater for the viewport, in each column of tiles or each row of tiles. As an MF store moves completely out of the viewport, it is relocated so as to start moving into the viewport from the opposite side and is updated with the appropriate image data.

The position of the screen in image space may be shifted by broadcasting an *x* and/or *y* offset to the pane coordinates of each MF store.

Various facilities may be provided to facilitate the processing of image data, such as copying data concurrently into a plurality of MF stores and copying the data from one such store to a plurality of other stores.

An example of repertoire of commands follows:

Micro-Frame Store Global Control:
- Set position of micro-frame store
- Associate micro-frame stores into arbitary windows
- Set a motion vector
- Set position of the screen in image space
- Write protect micro-frame store image memory
- Set priority of a micro-frame store
- Allocate a micro-frame store to colour channels.

Micro-frame Store Serial Communications:
- Copy data from off wafer to 'n' micro-frame stores
- Copy data from a micro-frame store to off wafer
- Copy data from a micro-frame store to 'n' micro-frame stores

where 'n' is any number from 1 to the maximum colour depth.

Micro-Frame Store Local I/O:
- Block Access
- Read all bits in access-array into data register
- Write all bits in access array from data register
- Set/Clear/Invert all selected* locations in the access array

- Row Access (if more than one row in the access array)
- Read any row into data register
- Write any row from data register
- Set/Clear/Invert all selected* locations in any row

*selection is achieved by 'setting' the corresponding bit in a mask register

Micro-Frame Store Video Output Stage:
- Enable/Disable Output
- Inverse/Normal Video
- Transparent/Opaque/Logical Overlays
- Force Output High/Low
- 2D Zoom
- Select a rectangular subsection of a micro-frame store for output

CLAIMS

1. A frame store assembly for a raster-scan display comprising a plurality of like bit-plane memory devices, each capable of storing pixel image data for a part area of the display, means for writing image data independently into the memory devices, and a pixel bus routed through each memory devices, each such device comprising programmable control means operable synchronously with respect to synchronising signals for the raster-scan display to read out on to the pixel bus selected data from the memory device at times which are determined by the programming of the memory device and thus cause the read-out data to provide

video data for a selected part area of the display, whereby the display is synthesized from the read-out data from selected memory devices.

2. A frame store assembly according to claim 1, wherein the programmable control means of each memory device comprise means for storing origin coordinates for the stored part area of the display, relative to a screen origin.

3. A frame store assembly according to claim 1 or 2, wherein the programmable control means of each memory device comprises means for storing coordinate values defining a sub-area of the stored part area and the programmable control means read out on to the pixel bus only the data pertaining to the said sub-area.

4. A frame store assembly according to claim 3, wherein the programmable control means of each memory device store origin coordinates for the sub-area relative to the origin of the part area, and extent coordinates for the sub-area relative to the origin of the sub-area.

5. A frame store assembly according to any of claims 1 to 4, wherein the programmable control means of each memory device include means for assigning the output of the device to different combinations of one or more lines of the pixel bus.

6. A frame store assembly according to any of claims 1 to 5, comprising a host processor and command and data bus means routed therefrom through the memory devices for writing image data in the memory devices and sending commands to their programmable control means.

7. A frame store assembly according to claim 6, wherein the pixel bus is routed through the memory devices in the reverse direction to the command and data bus means.

8. A frame store assembly according to claim 4 and claim 6 or 7, wherein the host processor is arranged to effect progressive changes in the horizontal and/or vertical coordinate direction of the part area origin coordinates and/or the sub-area origin coordinates to effect dragging, roaming, scrolling or panning of a viewport.

9. A frame store assembly according to claim 8, wherein the hose processor is arranged to associate a plurality of memory devices to define vertically or horizontally contiguous tiles of a viewport and to scroll or pan the image data therethrough by effecting like progressive changes in the vertical or horizontal part area origin coordinate and sub-area origin coordinate of all associated devices.

10. A frame store assembly according to claim 9, wherein the host processor associates at least one more memory device than the number required to cover the extent of the viewport and jumps the coordinates of a memory device whose image data has just moved out of the viewport so as to assign that memory device to the part of world space whose image data will move into the viewport on continued scrolling or panning.

11. A frame store assembly according to any of claims 1 to 10, wherein the pixel bus is accompanied through the memory devices by a line carrying the synchronizing signals from device to device.

12. A frame store according to claim 11, wherein the pixel bus is also accompanied through the memory devices by a priority bus routed through priority resolution units of the device.

13. A frame store according to claim 12, wherein the programmable control means of each memory device include means for storing a device priority number, each priority resolution unit is adapted to compare with the device priority number the incoming bus priority number on the priority bus, to pass on to the priority bus whichever number pertains to the higher priority and to control the introduction of read out data on to the pixel bus in dependence upon the sense of the priority resolution.

14. A frame store according to any of claims 1 to 13, wherein each memory device incudes a video output unit controlled by the programmable control means to select between different modes of introduction of data on to the pixel bus.

15. A frame store according to any of claims 1 to 11, wherein each memory device comprises a rectangular memory array storing pixel image data for a part area of the display which is a plurality of pixels wide and a plurality of pixels high.

16. A frame store according to any of claims 1 to 11, wherein each memory device comprises a linear memory array storing pixel image data for a part area of the display which is a plurality of pixels wide and one pixel high.

Printed for Her Majesty's Stationery Office by Croydon Printing Company (UK) Ltd, 1/87, D8817356.
Published by The Patent Office, 25 Southampton Buildings, London WC2A 1AY, from which copies may be obtained.

THIS PAGE BLANK (USPTO)